The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed [signature]

Dated 11 February 2004

Patents Act 1977
(Rule 16)

The Patent Office

03JUL03 E819688-1 D02246
P01/7700 0.00-0315508.2

THE PATENT OFFICE
D
-2 JUL 2003
LONDON

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference	P017500GB	
2. Patent application number *(The Patent Office will fill in this part)*	0315508.2	02 JUL 2003
3. Full name, address and postcode of the or of each applicant *(underline all surnames)*	ARM Limited 110 Fulbourn Road Cherry Hinton Cambridge CB1 9NJ, United Kingdom	
Patents ADP number *(if you know it)*		7498124002
If the applicant is a corporate body, give the country/state of its incorporation	United Kingdom	
4. Title of the invention	Power Control Within a Coherent Multi-Processing System	
5. Name of your agent *(if you have one)*	D Young & Co	
"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*	21 New Fetter Lane London EC4A 1DA	
Patents ADP number *(if you know it)*	59006	

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	0
Description	29
Claim *(s)*	4
Abstract	1
Drawing *(s)*	11 all

10. If you are also filing any of the following, state how many against each item.

Priority documents	0
Translations of priority documents	0
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	6
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	0
Any other documents *(please specify)*	0

11. I/We request the grant of a patent on the basis of this application.

Signature D Y & Co Date 02 July 2003

D Young & Co (Agents for the Applicants)

12. Name and daytime telephone number of person to contact in the United Kingdom — Nigel Robinson — 023 8071 9500

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

Patents Form 7/77

Patents Act 1977
(Rule 15)

The Patent Office

THE PATENT OFFICE
D
-2 JUL 2003
LONDON

Statement of inventorship and of right to grant of a patent

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1.	Your reference	P017500GB
2.	Patent application number *(if you know it)*	0315508.2 — 02 JUL 2003
3.	Full name of the or of each applicant	ARM Limited
4.	Title of the invention	Memory Control Within a Coherent Multi-Processing System
5.	State how the applicant*(s)* derived the right from the inventor*(s)* to be granted a patent	By Virtue of Employment
6.	How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))*	1
7.		I/We believe that the person*(s)* named over the page *(and on any extra copies of this form)* is/are the inventor*(s)* of the invention which the above patent application relates to. Signature [signature] Date 2 July 2003 D Young & Co (Agents for the Applicants)
8.	Name and daytime telephone number of person to contact in the United Kingdom	Nigel Robinson 023 8071 9500

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there are more than three inventors, please write the names and addresses of the other inventors on the back of another Patents Form 7/77 and attach it to this form.*

d) *When an application does not declare any priority, or declares priority from an earlier UK application, you must provide enough copies of this form so that the Patent Office can send one to each inventor who is not an applicant.*

e) *Once you have filled in the form you must remember to sign and date it.*

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Surname: PRUVOST

First Names: Julie-Anne Francoise Marie

41 avenue Reibaud
Residence ELISARA
06600 ANTIBES
France

Patents ADP number *(if you know it):* 8603854001

Surname: LATAILLE

First Names: Norbert Bernard Eugene

1099 Chemin de Saint Claude
Le Saint Claude B
06600 ANTIBES
France

Patents ADP number *(if you know it):* 8665846001

Surname: PIRY

First Names: Frederic Claude Marie

11 chemin des plateaux fleuris
06800 Cagnes sur Mer
France

Patents ADP number *(if you know it):* 8468324001

Reminder

Have you signed the form?

Patents Fc 7/77

Patents Act 1977
(Rule 15)

THE PATENT OFFICE
D
- 2 JUL 2003
LONDON

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

Statement of inventorship and of right to grant of a patent

1.	Your reference	P017500GB
2.	Patent application number *(if you know it)*	
3.	Full name of the or of each applicant	ARM Limited
4.	Title of the invention	Memory Control Within a Coherent Multi-Processing System
5.	State how the applicant*(s)* derived the right from the inventor*(s)* to be granted a patent	By Virtue of Employment
6.	How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))*	1
7.		I/We believe that the person*(s)* named over the page *(and on any extra copies of this form)* is/are the inventor*(s)* of the invention which the above patent application relates to. Signature Date 2 July 2003 D Young & Co (Agents for the Applicants)
8.	Name and daytime telephone number of person to contact in the United Kingdom	Nigel Robinson 023 8071 9500

Notes

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.

b) Write your answers in capital letters using black ink or you may type them.

c) If there are more than three inventors, please write the names and addresses of the other inventors on the back of another Patents Form 7/77 and attach it to this form.

d) When an application does not declare any priority, or declares priority from an earlier UK application, you must provide enough copies of this form so that the Patent Office can send one to each inventor who is not an applicant.

e) Once you have filled in the form you must remember to sign and date it.

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Surname: GRANDOU

First Names: Gilles Eric

13, rue Bourgarel Benoit
06130 Plascassier
France

Patents ADP number *(if you know it)*: 86663566001

Surname: GOODACRE

First Names: Anthony, John

8 Goose Cross
Highfields Caldecote
Cambridge
CB3 7ZL
United Kingdom

Patents ADP number *(if you know it)*: 86663331001

Surname:

First Names:

Patents ADP number *(if you know it)*:

Reminder

Have you signed the form?

POWER CONTROL WITHIN A COHERENT MULTI-PROCESSING SYSTEM

This invention relates to data processing systems. More particularly, this invention relates to data processing systems including multiple processor cores performing respective data processing operations and sharing access to a coherent memory region.

It is known to provide data processing systems including two or more processor cores which operate in a coherent multi-processing mode in which they share access to a coherent memory region. In such systems the different processor cores typically perform respective data processing operations in parallel to achieve an overall desired processing result.

An example of a coherent multi-processing system is the IBM370 system and SPARC multi-processor system. Such coherent multi-processing systems are high performance systems where power efficiency and power consumption is of little concern and the main objective is maximum processing speed.

An important consideration in coherent multi-processing systems is the management of coherency between cached copies of data values being held by different processor cores. It is known to provide memory access control units coupled to the processor cores which serve to perform coherency management operations to avoid situations such as a locally cached data value which is out-of-date being incorrectly used by a processor core when elsewhere within the coherent multi-processing system there is a more up-to-date version of that data value which should instead be used.

Viewed from one aspect the present invention provides apparatus for processing data, said apparatus comprising:

a plurality of processor cores operable to perform respective data processing operations, at least two of said processor cores being operable in a coherent multi-processing mode sharing access to a coherent memory region; and

a memory access control unit coupled to said plurality of processor cores and operable to perform coherency management operations with respect to at least one cached copy of a data value from within said coherent memory region; wherein

at least one of said processor cores operable in said coherent multi-processing mode is coupled to a cache memory, said cache memory being operable to remain active to service coherency management operations issued by said memory access control unit whilst said processor core coupled to said cache memory is in an inactive power saving state.

The invention recognises that within coherent multi-processing systems containing cached copies of a data value then advantageous power savings may be made whilst preserving the ability to maintain coherency by use of a technique whereby a processor core is powered down and made inactive whilst its cache memory storing the data values for which coherency needs to be maintained remains active and services coherency management operations generated by a memory access control unit without requiring the processor core itself to remain active. This technique runs counter to the normal practice in the field whereby a cache memory is typically powered down and rendered inactive when its associated processor core is powered down and rendered inactive. Maintaining the power to the cache has the advantages that power down of the core is speed up since there is no need to flush the cache, relatively fast access by other cores to the cached memory may be achieved avoiding relatively slow main memory accesses and upon wake up of the core there is a probability that required data will still be cached avoiding the need for a relatively slow refill.

A particularly convenient way of rendering the processor core inactive is to gate its clock.

It will be appreciated that the coherency management operations which need to be supported by the cache memory whilst the processor core is powered down can take a variety of different forms. In preferred embodiments of the present invention these

coherency management operations include a copy coherence management request to trigger return to the memory access management unit of a copy of a data value stored within the cache memory, a status change coherency management request from the memory access management unit serving to change a status value associated with a data value that is stored within the cache memory, and a clean coherency management request to trigger the cache memory to flush a dirty value stored therein to a main coherent memory.

Whilst it will be appreciated that the processor core advantageously saves power by being moved into its inactive state, it is important that it should be quick and easy to reactivate the processor core and accordingly preferred embodiments are ones in which the processor core is responsive to a received interrupt signal to return to the active powered state from the inactive power saving state.

Whilst it will be appreciated that the present technique may be advantageously used when only some of the processor cores have associated cache memories which remain active when their processor core is powered down, the invention is particularly suited for use in systems in which all of the processor cores have associated cache memories and all of these cache memories are ones which can remain active when their associated processing core is inactive.

Whilst it will be appreciated that the present technique may be embodied in a system in which the processor cores, cache memories, memory access control unit, etc are formed upon different integrated circuits or combinations of integrated circuits, the invention is particularly well suited when these elements are formed on a single integrated circuit.

Viewed from another aspect the present invention provides a method of processing data, said method comprising the steps of:

performing data processing operations upon respective ones of a plurality of processor cores, at least two of said processor cores being operable in a coherent multi-processing mode sharing access to a coherent memeory region; and

performing coherency management operations with respect to at least one cached copy of a data value from within said coherent memeory region using a memory access control unit coupled to said plurality of processor cores; wherein

at least one of said processor cores operable in said coherent multi-processing mode is coupled to a cache memory, said cache memory being operable to remain active to service coherency management operations issued by said memory access control unit whilst said processor core coupled to said cache memory is in an inactive power saving state.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates a data processing system including a plurality of processor cores;

Figure 2 schematically illustrates a memory bus between a processor core and a memory access control unit;

Figure 3 schematically illustrates a portion of an integrated circuit showing a processor core having a mode control parameter stored in the CP15 register;

Figure 4 schematically illustrates an integrated circuit having a mode control parameter stored in the memory control unit;

Figure 5 illustrates a processor core and a cache memory which are separately clocked such that the processor core may be powered down whilst the cache memory remains responsive to coherency management operations; and

Figures 6 to 11 illustrate further details of a multi-processor architecture and bus interface in accordance with example embodiments of the present techniques.

Figure 1 schematically illustrates an integrated circuit 2 containing a plurality of microprocessor cores 4, 6, 8, each with an associated cache memory 10, 12, 14. The processor cores 4, 6, 8 are connected by respective memory buses AHB, CCB to a memory management access unit 16 (also called a snoop control unit). A peripheral device 18 is provided as a private peripheral connected to one of the processor cores 4.

The integrated circuit 2 is coupled to a memory 20 by one of several possible master AHB ports. The memory 20 contains a coherent shared region 22. Memory may be configured and used as non-coherent shared memory when more than one processor has access to it, e.g. a general purpose processor core and a specialist DSP core may share access to a common memory region with no control of coherency being performed. Coherent shared memory is distinguished from non-coherent shared memory in that in coherent shared memory the mechanisms by which that memory is accessed and managed are such as to ensure that a write or a read to a memory location within that coherent shared region will act upon or return the current and most up-to-date version of the data value concerned. Thus, coherent shared memory is such that if one processor core makes a change to a data value within the coherent shared region, then another processor core will read that up-to-date data value when it seeks to access that data value. Furthermore, a write to a data value within the coherent memory region 22 will force a change in other stored copies of that data value, at least to the level of ensuring that out-of-date copies are marked as invalid and so subsequently not used inappropriately.

In the system of Figure 1, the snoop control unit 16 is responsible for managing access to the memory 20, and the coherent shared memory region 22 in particular. The

snoop control unit 16 keeps track of which processor cores 4, 6 that are acting in a coherent multi-processing mode are currently holding local copies of a data value from the coherent memory region 22 within their cache memories 10, 12. Coherency management is in itself a known technique. Descriptions of such techniques may be found for example within the Western Research Laboratory Research Report 95/7 entitled "Share Memory Consistency Models: A Tutorial" by Sarita V. Adve and Kourosh Gharachorloo; University of Wisconsin – Madison Computer Sciences Technical Report/902 December 1989; "Week Ordering – A New Definition And Some Indications" by Sarita V. Adve and Mark D Hill; and "An Implementation Of Multi Processor Linux" by Alan Cox, 1995. Whilst coherent multi-processing itself is an established technique, the provision of such capability with reduced hardware complexity overhead, backward compatibility and configuration flexibility is a significant challenge.

Figure 2 illustrates the memory bus between the processor cores 4, 6, 8 and the snoop control unit 16 in more detail. In particular, this memory bus is formed of an AHB bus (AMBA High-Performance Bus) in parallel with a coherency control bus (CCB). The AHB bus has the standard form as is known from and described in documentation produced by ARM Limited of Cambridge, England. This AHB bus is a uni-processing bus with the normal capabilities of operating with processor cores performing uni-processing (or non-coherent multi-processing such as a core and a DSP accessing a shared non-coherent memory). The AHB bus does not provide capabilities for coherent multi-processing. Private peripheral devices, such as a peripheral device 18 as illustrated in Figure 1, may be connected to this bus without modification providing they do not need to access the coherent multi-processing capabilities of the system. This provides advantageous backward compatibility with existing peripheral designs.

The coherency control bus CCB can be considered to provide a number of respective channels of communication between the attached processor core 4, 6 and the snoop control unit 16. In particular, the core may generate coherency request signals, core status signals and core side band signals which are passed from the

processor core 4, 6 to the snoop control unit 16. The snoop control unit 16 can generate coherency commands that are passed from the snoop control unit 16 to the respective processor core 4, 6.

The CCB in particular is used to augment signal values on the AHB to provide additional information from the core 4, 6 to the snoop control unit 16 characterising the nature of a memory access being requested such that the coherency implications associated with that memory access request can be handled by the snoop control unit 16. As an example, line fill read requests for the cache memory 10, 12 associated with a coherent multi-processing core 4, 6 may be augmented to indicate whether they are a simple line fill request or a line fill and invalidate request whereby the snoop control unit 16 should invalidate other copies of the data value concerned which are held elsewhere. In a similar way, different types of write request may be distinguished between by the coherency request signals on the CCB in a manner which can then be acted upon by the snoop control unit 16.

The core status signals pass coherency related information from the core to the snoop control unit such as, for example, signals indicating whether or not a particular core is operating in a coherent multi-processing mode, is ready to receive a coherency command from the snoop control unit 16, and does or does not have a data value which is being requested from it by the snoop control unit 16. The core sideband signals passed from the core to the snoop control unit 16 via the CCB include signals indicating that the data being sent by the core is current valid data and can be sampled, that the data being sent is "dirty" and needs to be written back to its main stored location, and elsewhere as appropriate, that the data concerned is within an eviction write buffer and is no longer present within the cache memory of the core concerned, and other signals as may be required. The snoop control unit coherency commands passed from the snoop control unit 16 to the processor core 4, 6 include command specifying operations relating to coherency management which are required to be performed by the processor core 4, 6 under instruction of the snoop control unit 16. As an example, a forced change in the status value associated with a data value being held within a cache memory 10, 12 of a processor core 4, 6 may be instructed such as

to change that status from modified or exclusive status to invalid or shared in accordance with the applied coherency protocol. Other commands may instruct the processor core 4, 6 to provide a copy of a current data value to the snoop control unit 16 such that this may be forwarded to another processor core to service a memory read request, from that processor core. Other commands include, for example, a clean command.

Figure 3 illustrates a section of an integrated circuit 2 according to an embodiment of the invention. The integrated circuit 2, comprises a memory access control unit 16, (often referred to as the snoop control unit or memory management access unit), a memory 20 and a plurality of processor cores 4, 6. The processor cores include processor core 4 that is configurable to operate either in non-coherent processing mode or in coherent multi-processing mode. The other processor cores (not all shown in Figure 3) may be multi-processor cores, non-coherent processor cores or they may be like processor core 4 configurable to operate as either.

Processor cores operating in coherent multi-processing mode have access to a shared memory region, this region being cachable by the cores operating in coherent multi-processing mode and a defined portion of memory 20. Processor cores operating in non-coherent mode do not access coherent shared memory region and their caches do not mirror any data contained in these regions.

Although memory 20 is shown as a block on the integrated circuit 2, this is purely for ease of illustration and in reality memory 20 may include a variety of data stores on and/or off the integrated circuit and also the caches of the processor cores.

Processor core 4 has an associated cache memory 10 and a mode control parameter storage element, which in this embodiment is part of the CP15 register. The mode control parameter controls the processor core to operate either in non-coherent processing mode or in coherent multi-processing mode. The parameter may be set in a variety of ways including in response to a software command from an application or

operating system, or it may be hardware controlled by a signal on an external pin 11.

As in the other embodiments processor core 4 communicates with the snoop control unit via a bus. This bus is divided into two portions, the main or AHB portion and the multi-processing or CCB (coherency control bus) portion. The main portion is used to transmit memory access signals from the processor core to the snoop control unit and from the snoop control unit to the core, the additional portion is used for additional information related to coherency management operations.

In operation when the mode control parameter is set to indicate that the processor core is to operate in non-coherent processing mode, the core acts in response to this signal to de-activate the CCB. This means that memory access signals are sent by the AHB bus alone and have no additional coherency related data attached to them. As no additional coherency information is received by the snoop control unit 16 it performs no coherency operations on the memory access request but simply directs the memory access request to the relevant portion of memory 20.

As can be seen from Figure 3, in addition to controlling the core 4 to de-activate the CCB, the mode control parameter is sent directly to the snoop control unit 16 as an SMP/AMP signal. As in this case the mode control parameter is set to indicate that the processor core 4 is operating in non-coherent processing mode, the signal received by the snoop control unit 16 indicates that the cache 10 of processor core 4 is not mirroring any shared memory. Cache memory 10 is therefore not relevant to the snoop control unit 16 when it is servicing memory access requests from other cores and the snoop control unit 16 therefore ignores cache memory 10 when servicing memory access requests from other processor cores.

When the mode control parameter is set to indicate that processor core 4 is to operate in coherent multi-processing mode, the CCB bus is not automatically de-activated. In this circumstance the core may produce additional information to describe a particular memory access request and act to transmit the memory access request on the AHB bus and the additional data on the CCB bus. The receipt of the

additional information on the CCB bus indicates to the snoop control unit that processor core 4 is operating in coherent multi-processing mode and that coherency management operations need to be performed. In some circumstances the memory access request is such that although the core is operating in coherent multi-processing mode it knows that there are no coherency problems associated with this particular request. In these circumstances, for example, where the core knows that the latest version of the data it needs to read is in its own cache, the core acts to de-activate the CCB as in the non-coherent processor mode and no additional information is sent with the memory access request. In this case as in the non-coherent processing mode example the snoop control unit knows that no coherency management operations need to be performed and thus it simply directs the memory access request to the memory location indicated.

As in this case the mode control parameter is set to indicate coherent multi-processing mode, the cache 10 of processor core 4 mirrors part of the shared memory accessible to other processor cores 6 operating in coherent multi-processing mode and is thus relevant to the snoop control unit 16 servicing memory access requests from coherent multi-processing mode processors. As the snoop control unit 16 receives a signal giving the value of the mode control parameter it is aware of this and as such does not ignore the cache 10 of core 4 when servicing memory access requests from other processor cores operating in coherent multi-processing mode.

Figure 4 shows an alternative embodiment where the processor cores 4, 6, 8 are all configurable to operate either in multi-processing or in non-coherent processing mode. In this embodiment the mode control parameters are not stored on the processor cores themselves but are rather stored on the snoop control unit 16. In the embodiment shown these signals are sent to the cores and can be used by the cores, as in the embodiment illustrated in Figure 3, to disable the CCB if they indicate the processor core to be operating in non-coherent processor mode. As they are stored on the snoop control unit 16, the snoop control unit has access to them and uses them to determine which processor core caches it needs to access when servicing memory access requests from coherent multi-processing mode processor cores.

Although the two embodiments illustrated have shown the control parameters stored either in the configurable core 4 or on the snoop control unit 16, it would be possible to store these parameters elsewhere on the integrated circuit 2. In all of these embodiments the control parameters may be set in a variety of ways including in response to a software command from an application or operating system, or they may be hardware controlled by a signal on an external pin (not shown).

Figure 5 schematically illustrates a processor core 4 with an attached cache memory 10. This cache memory 10 is a 4-way physically addressed cache memory. The cache memory 10 is supplied with its own clock signal. The clock signal which is supplied to the processor 4 may be gated off by a control gate 24 whilst the clock continues to be supplied to the cache memory 10. Thus, the processor core 4 may be stopped and placed into a power saving mode by gating off its clock with the control gate 24. A status flag within a core configuration coprocessor CP15 is used to switch the control gate 24 between allowing the clock signal to pass and gating off the clock signal. One type of WFI (wait for interrupt) instruction is used to trigger setting of this status flag and gating of the core clock while the cache clock remains active. Another type of WFI instruction may be used to gate the clock to both the core and the cache.

Within the cache memory 10, a coherency command decoder 26 is provided and is responsive to coherency commands passed via the CCB from the snoop control unit 16. These coherency commands include forcing a change in status associated with a data value held within the cache memory 10, returning a copy of a data value held or cleaning a data value held as instructed by the snoop control unit 16. Thus, whilst the processor core 4 may be placed into a power saving mode to reduce overall system power consumption, the cache memory 10 can remain responsive to coherency management requests issued by the snoop control unit 16 and directed to it via the CCB. This enables significant power saving whilst not compromising the coherency management.

A further description of the multi-processor architecture in general is given in the following:

Terms and abbreviations

This document uses the following terms and abbreviations.

Term	Meaning
SMP	Symmetric Multi-Processing
AMP	Asymmetric Multi-Processing
L2CC	Level Two Cache Controller
WFI	Wait For Interrupt. Low power mode. All clocks in the core are switched off, the core being awaken on the receipt of an interrupt.

Introduction

We describe hereafter a global Multi-processing platform. The specified architecture should allow both SMP and AMP within the same platform, with the same programmer's model.

A typical MP system includes:

- Memory coherency support;
- Interrupt distribution mechanism;
- Inter-processor communication channels;
- Multi-core debug capabilities;
- Multi-core trace capabilities.

This architecture enables the development of Low Power Multi-processing systems (the WFI state for Low Power mode is supported).

This architecture should scale to cores having a private Level 2 cache.

Ease of integration of this architecture into already existent designs has been considered. The current specification should allow replacing a single core with an SMP-capable system with no other change in the design.

SMP SOLUTION

COHERENT MULTIPROCESSING MEMORY SYSTEM

The chosen solution is shown in Figure 1:

Two main tasks were identified to produce a multi-processing memory system:

- Add MP extensions to the ARM core to produce a Multiprocessor-capable core. These modifications include moving the core to physical addressing, updating the cache line states, and adding a Coherency Control Bus (CCB) at core interface ;
- Produce a block responsible for the memory system coherency, dubbed the Snoop Controller Unit (SCU). This block implements the MESI coherency protocol at the system level and sends coherency requests to cores in the memory system.

SMP-capable cores

Standard ARM cores should be modified to take advantage of the Multi-Processing environment:

- They can and receive messages to/from the Snoop Control Unit (SCU) through the Coherency Control Bus (CCB) ;
- They handle SMP information in their cache lines, like basic MESI states, SMP/AMP awareness and migratory-lines detection ;
- They may provide new MP instructions, to support a better locking mechanism.

However, an important point is that an SMP capable core will still be compatible with the standard AHB bus, and can work seamlessly in a non-Multiprocessing memory environment.

The Snoop Controller Unit

In the ARM MP-architecture, a centralized unit (dubbed the SCU, for Snoop Control Unit) controls AHB requests coming from the cores and checks them for coherency needs. This unit ensures that memory consistency is maintained between all caches. When necessary it sends control messages to data caches (INVALIDATE, CLEAN or COPY commands) and redirects memory transfers (directly between processors, or to the external AHB interface).

Different features can be added to the SCU. These features are mostly transparent to the programmer, and can improve performance and/or power consumption. These may be configurable, and can be arranged to ensure that their default configuration does not change the programmer's model. Although this is not mandatory, the SCU can for example maintain a local copy of all processors DATA TAG arrays to speed-up coherency lookups without having to ask (and therefore stall) processors in the memory system.

The SCU also uses an external master AHB interface. This interface can send writes requests to memory, and read data from the main memory if the requested line is not present in other Data caches (snoop miss). In order to ease the implementation of a SMP-capable system, this external interface is designed to plug easily to a L2CC, an AMBA3 wrapper or a standard AHB bus.

COHERENT PROTOCOL AND BUSSES

Snooping activity and coherency protocol

At the SCU level, each memory request coming from an SMP core generates a coherency check. Only data-side caches of processors in the SMP memory system are looked up for the required data.

The cache coherency protocol used for the Core-SCU communication is based on the MESI protocol. However, it has been modified using a Berkeley approach to improve its performance and power consumption.

In a Multiprocessing memory system, the *consistency model* is an important piece of the Programmer's model. It defines how the programmer should expect the memory content to change while issuing reads and writes. The consistency model of the ARM MP architecture is the Weak Ordering model, which ensures correct program behaviour using synchronisation operations.

Coherency Control Bus

A bus between the core and the SCU, dubbed the Coherency Control Bus (CCB), is responsible for passing messages between the SCU and the cores. This defines a standard interface between a SMP capable core and the SCU.

As the SMP architecture evolves this allows the SMP-core interface to remain stable.

This bus is also providing status signals mandatory to implement Multiprocessing features, as described in the *Supported Features* section given below.

SUPPORTED FEATURES

SMP/AMP attribute

In a multiprocessor system, one could imagine dedicating one or more processor(s) to non-SMP tasks / OS. This means that this (these) processor(s) will never handle shared data.

This can be the case if someone wants to avoid porting applications from one OS to a new one. The solution is to run a separate OS on a dedicated processor, even if this OS is not SMP capable. This can also be considered for specific tasks/threads that do not need any OS support, like for example when running a dedicated multimedia task on a separate processor (which may have a specific or private coprocessor).

Processing coherency checks on each AHB request from these processors is useless, since they will never share data, and it penalises the performances of both the whole system (since you will add load to the SCU) and the processor itself (since you introduce latency on the AHB request for looking for coherency needs).

An attribute in CP15 defines whether the processor is working in symmetrical mode or not. It defines if AHB requests from the processor should be taken into account by the SCU and whether this processor's Data cache has to be looked at upon coherency requests from other processors.

This attribute is sent to the SCU as a SCSMPnAMP bit.

Direct Data Intervention

Description

When a processor requires a line which is stored in another processor's cache, the SCU can transmit the line from the processor having it to the one requesting it.

The goal is to limit accesses to the following memory level, those accesses penalising both timing and power consumption. The SCU will hence get the line from the owner, and will forward it to the requiring processor.

Different line status changes are defined, depending on the state of the line in the owning processor (Modified, Shared or Exclusive), the type of request (read or write) and whether the migratory line feature is enabled or not.

Coherency with core OFF and caches ON

An additional Wait-for-Interrupt instruction has been defined that allows turning off the core while maintaining coherency in the L1 caches (caches ON).

MP-capable cores thus have two Wait-for-interrupt instructions:

- A WFI instruction that puts both the core and the caches in a low-power state.
- A WFI instruction that puts the core in a low-power state while the caches are still ON and able to service coherency requests from the SCU (FORCE/COPY and CLEAN operations)

Both WFI instructions are implemented as CP15 instructions.

The way the low-power state is achieved is through clock-gating. A module at the CPU level stops the clock of the core or the clock of both the core and the cache.

The core escapes the low-power WFI state upon reception of an interrupt.

The Coherency Control Bus (CCB)

The Coherency Control Bus (CCB) is responsible for passing coherency messages between an ARM MP-capable core and the Snoop Controller Unit (SCU).

This bus is originally designed for a multi-processing system based on the ARM1026 core family. The AMBA bus used between the ARM1026 core and the SCU is a private one.

However, the defined CCB specification is also applicable to the following memory environments:

- AHB-lite memory systems (using multiple private slaves at the core level) ;
- Full AHB memory systems (featuring multiple masters at the core level) ;
- AXI memory systems (AHB 3.0) with minor modifications.

The bullet specification of this Coherent Control Bus (CCB) is:

- Sideband signals are added to the AMBA bus at the master interface, on control and data paths ;
- Coherent AMBA requests (requests with the SCREQ sideband signal asserted) must be dispatched to the Snoop Control Unit ;
- The Snoop Controller Unit uses a private channel to send coherency commands to the core ;
- Requested coherent data and core notification messages are sent to the SCU as AMBA write accesses ;

In the following chapter, we present the CCB scheme with more details in an AHB 2.0 memory environment.

CCB OVERVIEW

Sideband signals on core requests

When sending a memory request on the AMBA bus, a Multi-Processing aware core sets the “CCB core sideband” signals to indicate what type of memory burst is needed.

The value of this sideband bus distinguishes between the following operations:

- standard Read and Write AMBA requests ;
- coherent “Line Fill” and “Line Fill and Invalidate” read requests ;
- coherent “Write Through and Invalidate”, “Write Not Allocate and Invalidate” and “Invalidate” write requests ;
- “CP15 Invalidate” and “CP15 Invalidate All” notifications ;
- requested “CLEAN / COPY data transfers” ;

A precise list of signals with their encoding is available below.

SCU coherency command channel

While ensuring the memory system consistency, the SCU may have to send coherency commands to all cores in the memory system.

The following coherency operations are defined:

- change the state of a cache line (FORCE command) ;
- change the state of a cache line and CLEAN the line contents on the bus ;
- change the state of a cache line and COPY the line contents on the bus ;
- do nothing (NOP command).

Together with the coherency operation, a MESI state is sent. It indicates the final state of the cache line once the coherency operation has been processed.

The Snoop Controller Unit uses a private communication channel to send coherency commands to the core:

- the SCOP bus indicates to the core which coherency operation is needed ;
- the SCCOREREADY signal indicates to the SCU if the current coherency request has completed, and if the core is ready to process another request (in a similar way to the AHB HREADY signal).

This bus does not depend on the AMBA bus. If a coherency request is required by the SCU while the SCCOREREADY signal is asserted, the core has to register the coherency request and drop the SCCOREREADY signal.

The SCCOREREADY signal should remain LOW as long as the core has not completed the coherency operation.

Please refer to timing diagrams and description below for more information regarding coherency requests management.

Sending CP15 notifications

When a core issues a "CP15 INVALIDATE" or "CP15 INVALIDATE ALL" command on its data cache, it has to send a message to the SCU unit. This message is needed to force the SCU to update its Dual Tag arrays.

This "CP15 notification" message is sent by the core as a single AHB WRITE cycle as follows (see timing diagrams):

- SCREQ = HIGH, indicating a coherent request addressed to the SCU block ;
- SCINV = LOW and SCDATA = LOW, indicating a "CP15 INVALIDATION" notification message ;
- The WDATA bus value is not relevant for this message. At the SCU level, this request is considered as "CP15 notification", and thus will not be forwarded to main memory ;
- The HADDR bus value is not relevant for this memory access. Instead this bus contains the Index+Way address for the invalidation operation.

This means that the AMBA address decoding logic (if any) sitting between the core and the SCU should always select the SCU slave port when receiving a memory request which has the SCREQ bit asserted.

Processing coherency requests at the core level

When the core receives a coherency command coming from the SCU on the SCOP bus, it registers the requested operation and is getting prepared to service the request.

Many cases may appear at the core interface:

a) If the core is not processing any memory transfer at the BIU interface, it can start the coherency request immediately (FORCE / CLEAN / COPY).

If cleaned / copied data must be sent back to the SCU, the core produces an incrementing AMBA WRITE burst as follows (see timing diagrams below):

- SCREQ = HIGH, indicating a coherent request addressed to the SCU block ;
- SCINV = LOW and SCDATA = HIGH, indicating a "COPY/CLEAN transfer" ;
- The SCDATAVALID and SCDIRTYWR are updated on a data basis ;

- As for CP15 notification messages, the HADDR sent value is not relevant for this message.

At the SCU level, this message is considered as a "COPY/CLEAN transfer" and will not be forwarded to main memory.

b) If the core is processing / requesting a non-coherent data (SCREQ signal is not asserted), it can complete his current burst as usual. This is the case when the core is processing either a memory transfer to a private slave or a non-coherent memory transfer to the SCU.

Once the burst has completed, the core must then process the "CLEAN or COPY data transfer" as explained in case a/.

c) If the core is processing / requesting a memory request (SCREQ signal is asserted), this means that the core is currently issuing a coherent memory transfer with the SCU.

In this case, the transfer cannot complete until the core has serviced the coherency command sent by the SCU. The reason for this behaviour is that it may hide a deadlock case for the memory system.

It is guaranteed that the SCU will not process the stalled request further (by asserting HREADY to HIGH or sending data back) until the coherency command has been serviced. The core must start processing the coherency request (FORCE / CLEAN / COPY).

If cleaned / copied data must be sent back to the SCU, the core can send it to the SCU on the WDATA bus while setting SCDATAVALID and SCDIRTYWR signals on a data basis (see timing diagrams below).

CCB SIGNALS

The Coherency Control Bus (CCB) can be divided in 4 signal groups:

- Core coherency request signals: these signals are controlled by the core and are sent in parallel with the AMBA request. They indicate if the AMBA request is a coherent one, and tell the SCU what kind of coherency action is required.

  The following coherent memory requests are defined:

  - LF [Line fill]: issued when a read miss happens in a processor. This command requests a line in either shared or exclusive state. The final state will depend on the SCU's answer.
  - LFI [Line Fill and Invalidate]: issued when a write miss happens in a processor, if Write Allocation is enabled. This command requests exclusive ownership of a line.
  - WTI [Write Through Invalidate]: issued when the cache is configured in Write Through mode. In this case, the SCU must invalidate the corresponding line in other processors if needed. In the case where the processor has already the line either in Exclusive or Modified state, the command will not be issued.
  - WNAI [Write non-allocate invalidate]: Issued when the cache is configured in write non-allocate mode, and the line isn't in the cache. The SCU must then invalidate the line in other processors if needed.
  - Invalidate: issued on a Write Hit to the cache, with the line being in shared state. We do not need to send data on the bus. Upon reception of this message, the SCU invalidates lines in other caches.
  - CP15 invalidations: those messages are used to update the DUAL TAG ARRAYS located in the SCU.

- Core status signals: these signals are coherency status signals sent by the core. They indicate if the core is ready to process coherency commands coming from the SCU, and they give the status of the current coherency request.
- Core sideband signals: these signals are sent by the Core in parallel with the data during a coherency operation.

- SCU command signals: these signals are used by the SCU to send coherency commands to the core.

Core coherency request signals (in parallel with AHB request)			
Name	**Width**	**Output**	**Description**
SCREQ	1 bit	Core	Indicates that the AMBA request must be checked for coherency. It remains stable for the duration of the request. SCREQ must always be equal to zero if SCSMPnAMP is clear or if the request is not addressed to the SCU. • SCREQ = 1'b0: normal AHB reads and writes – no coherency check is performed. • SCREQ = 1'b1: the current request is a coherent request/message addressed to the SCU.
SCINV	1 bit	Core	Together with the type of the AHB transaction and SCREQ, distinguishes between: • LF (0) and LFI (1) requests ; • CP15 operations or coherent COPY/CLEAN DATA TRANSFERS (0) and WTI / WNAI or INVALIDATION (1) requests ; This signal is stable during a memory request.
SCWT	1 bit	Core	Together with the type of the AHB transaction and SCREQ, distinguishes

			between: • WNAI (0) and WTI (1) requests. This signal is stable during a memory request.
SCALL	1 bit	Core	Together with the type of the AHB transaction and SCREQ, distinguishes between: • CP15 INVALIDATE (0) and CP15 INVALIDATE ALL (1) requests. This signal is stable during a memory request.
SCDATA	1 bit	Core	Together with the type of the AHB transaction and SCREQ, distinguishes between: • INVALIDATE (0) and WTI / WNAI (1) requests • CP15 INVALIDATE / CP15 INVALIDATE ALL operations (0) and coherent COPY/CLEAN DATA TRANSFERS (1)
SCWAY	4 bits	Core	Indicates which cache way is used by the core for the current Line Fill request. It is also used with the "CP15 INVALIDATE ALL" message to indicate which ways are to be cleaned. This signal is encoded using 1 bit per cache way.

Core status signals

Name	Width	Output	Description
SCSMPnAMP	1 bit	Core	Indicates whether or not the processor is part of the SMP system, i.e. if this processor's Data cache has to be looked at upon coherency requests from other processors. When clear, the processor is totally isolated from the MP cluster and is not part of the snooping process. The Dual Tag array information is not maintained for this processor. The SCSMPnAMP value can be changed at the core level through a CP15 operation. It must remain stable when a memory request is being processed.
SCCOREREA DY	1 bit	Core	Indicates that the core is ready to receive a coherency request from the SCU (See timing diagrams below).
SCnPRESEN T	1 bit	Core	Not Present bit: indicates that the line requested by the SCU is no longer present in the core's cache. This signal is valid in the cycle when SCCOREADY indicates the completion of the request (See timing diagrams below).

Core sideband signals			
Name	**Width**	**Output**	**Description**
SCDATAVALI D	1 bit	Core	Indicates that the data sent by the core is valid and can be sampled (See timing diagrams below).

SCDIRTYWR	1 bit	Core	Dirty attribute sent along with the data for COPY and CLEAN coherency operations (See timing diagrams below).
SCEWBUPDATE	1 bit	Core	Indicates that a data line has been placed in the Eviction Write Buffer in core and is not present in the data RAM. Valid on cache Line Fills, and in the first cycle of a "CP15 INVALIDATE" message (See timing diagrams below).

SCU command signals			
Name	**Width**	**Output**	**Description**
SCOP	2 bits	SCU	Coherency operation sent by the SCU to the core: • "00" : NOP • "01" : FORCE cache line state value • "10" : COPY • "11" : CLEAN
SCUMIG	1 bit	SCU	Indicates that the incoming cache line is migratory so that the Cache State Machine can react accordingly (optional signal).
SCADDR	32 bits	SCU	Snooping Address bus This bus is used to send coherency requests to a core. It can hold a Physical Address, an Index/Way value, or a direct link to the core's Eviction Write Buffer.
SCSTATE	2 bits	SCU	Indicates the final cache line state after a coherency operation or a "Line Fill" / "Line

			Fill Invalidate" request (See timing diagrams): • "00" : Invalid • "01" : Shared • "10" : Exclusive • "11" : Modified

SCREQ	HWRITE	SCINV	SCDATA	SCWT	SCALL	Coherency message
0	-	-	-	-	-	Standard memory request
1	0	0	-	-	-	Line Fill request
1	0	1	-	-	-	Line Fill and Invalidate request
1	1	0	0	-	0	CP15 INVALIDATE request
1	1	0	0	-	1	CP15 INVALIDATE ALL request
1	1	0	1	-	-	coherent CLEAN / COPY transfer
1	1	1	0	-	-	INVALIDATE request

1	1	1	1	0	-	WNAI request
1	1	1	1	1	-	WTI request

Coherency messages encoding (Core to SCU)

AHB2.0 TIMING DIAGRAMS

The following timing diagrams explain the core / SCU communication :

- Line Fill example ;
- Invalidate All example ;
- FORCE command example (Not Present case) ;
- COPY command example (hit case) ;
- CLEAN command example (miss case) ;
- Coherent write burst delayed by a COPY command.

Coherent Line Fill request

(See Figure 6)

INVALIDATE ALL message

(See Figure 7)

FORCE command (not Present Case)

(See Figure 8)

COPY command (hit case)

(See Figure 9)

CLEAN Command (miss case)

(See Figure 10)

Coherent write burst delayed by a COPY command

(See Figure 11)

CLAIMS

1. Apparatus for processing data, said apparatus comprising:

a plurality of processor cores operable to perform respective data processing operations, at least two of said processor cores being operable in a coherent multi-processing mode sharing access to a coherent memory region; and

a memory access control unit coupled to said plurality of processor cores and operable to perform coherency management operations with respect to at least one cached copy of a data value from within said coherent memory region; wherein

at least one of said processor cores operable in said coherent multi-processing mode is coupled to a cache memory, said cache memory being operable to remain active to service coherency management operations issued by said memory access control unit whilst said processor core coupled to said cache memory is in an inactive power saving state.

2. Apparatus as claimed in claim 1, wherein said processor core coupled to said cache memory is not clocked in said inactive power saving state.

3. Apparatus as claimed in any one of claims 1 and 2, wherein said cache memory is responsive to a copy coherency management request received from said memory access management unit to return a copy of a data value stored within said cache memory.

4. Apparatus as claimed in any one of claims 1, 2 and 3, wherein said cache memory is responsive to a status change coherency management request received from said memory access management unit to change a status value associated with a data value stored within said cache memory.

5. Apparatus as claimed in any one of the preceding claims, wherein said cache memory is responsive to a clean coherency management request received from said

memory access management unit for a value stored within said cache memory to, if said value is dirty, then to return said dirty data value to a main memory.

6. Apparatus as claimed in any one of the preceding claims, wherein in said inactive power saving state said processor core is responsive to a received interrupt signal to return to an active powered state.

7. Apparatus as claimed in any one of the preceding claims, wherein a wait for interrupt instruction executed by said apparatus triggers said processor core to enter said inactive power saving state whilst said cache memory remains in said active state.

8. Apparatus as claimed in any one of the preceding claims, wherein each of said processor core operable in said coherent multi-processing mode is coupled to a respective cache memory.

9. Apparatus as claimed in any one of the preceding claims, wherein said apparatus comprises an integrated circuit including said plurality of processor cores, said memory access control unit and said cache memory.

10. A method of processing data, said method comprising the steps of:

performing data processing operations upon respective ones of a plurality of processor cores, at least two of said processor cores being operable in a coherent multi-processing mode sharing access to a coherent memory region; and

performing coherency management operations with respect to at least one cached copy of a data value from within said coherent memeory region using a memory access control unit coupled to said plurality of processor cores; wherein

at least one of said processor cores operable in said coherent multi-processing mode is coupled to a cache memory, said cache memory being operable to remain active to service coherency management operations issued by said memory access

control unit whilst said processor core coupled to said cache memory is in an inactive power saving state.

11. A method as claimed in claim 10, wherein said processor core coupled to said cache memory is not clocked in said inactive power saving state.

12. A method as claimed in any one of claims 10 and 11, wherein said cache memory is responsive to a copy coherency management request received from said memory access management unit to return a copy of a data value stored within said cache memory.

13. A method as claimed in any one of claims 10, 11 and 12, wherein said cache memory is responsive to a status change coherency management request received from said memory access management unit to change a status value associated with a data value stored within said cache memory.

14. A method as claimed in any one of claims 10 to 13, wherein said cache memory is responsive to a clean coherency management request received from said memory access management unit for a value stored within said cache memory to, if said value is dirty, then to return said dirty data value to a main memory.

15. A method as claimed in any one of claims 10 to 14, wherein in said inactive power saving state said processor core is responsive to a received interrupt signal to return to an active powered state.

16. A method as claimed in any one of claims 10 to 15, wherein execution of a wait for interrupt triggers said processor core to enter said inactive power saving state while said cache memory remains in said active state.

17. A method as claimed in any one of claims 10 to 16, wherein each of said processor core operable in said coherent multi-processing mode is coupled to a respective cache memory.

18. A method as claimed in any one of claims 10 to 17, wherein said plurality of processor cores, said memory access control unit and said cache memory are part of a single integrated circuit.

ABSTRACT

POWER CONTROL WITHIN A COHERENT MULTI-PROCESSING SYSTEM

Within a multi-processing system including a plurality of processor cores 4, 6 operating in accordance with coherent multi-processing, each of the cores includes a cache memory 10, 12 storing local copies of data values from a coherent memory region. The respective processor cores may be placed into a power saving mode in which they are non-operative whilst the cache memory remains responsive to coherency management requests such that the system as a whole can continue to operate and manage coherency.

[Figure 5]

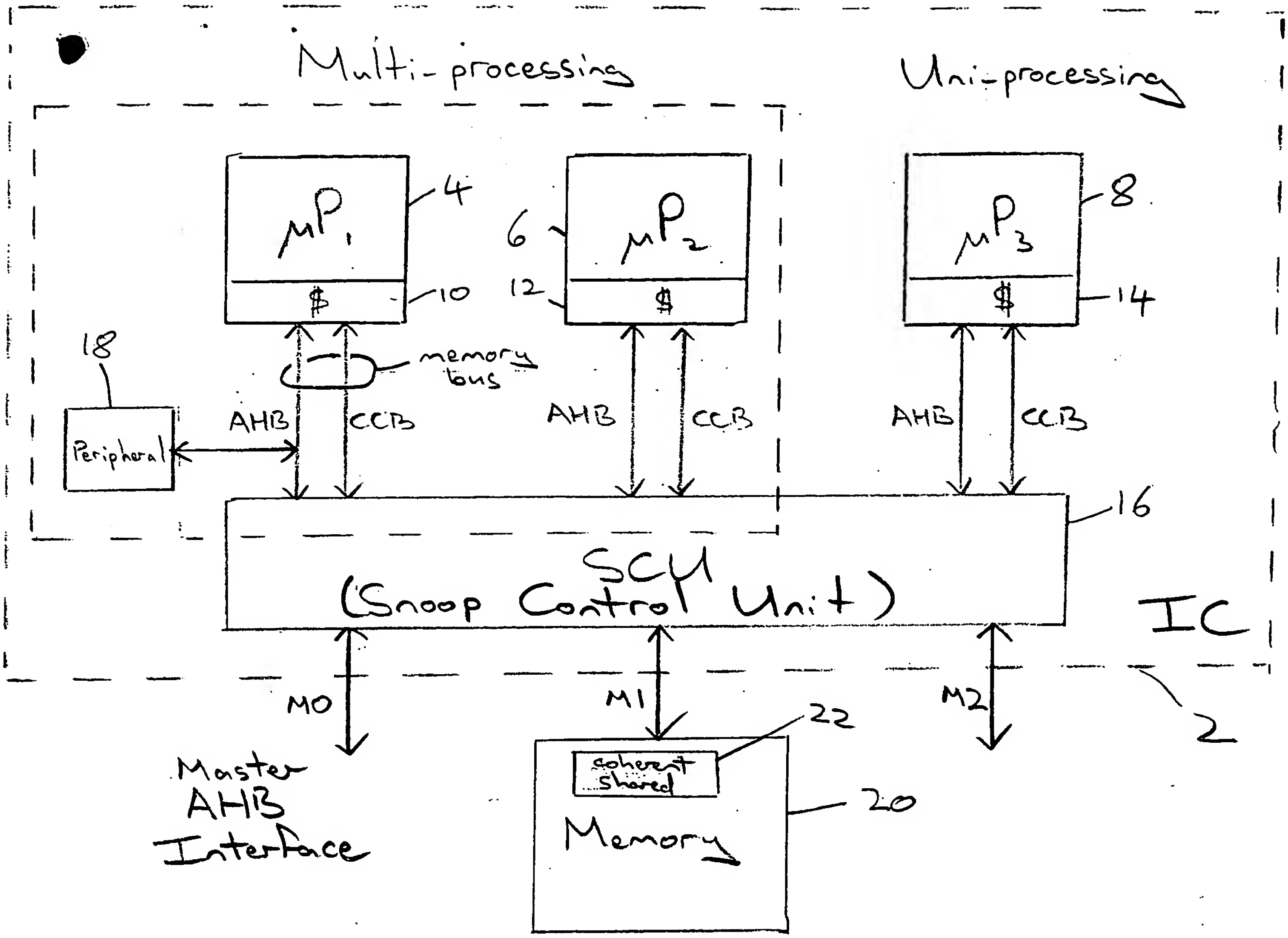
Multi-processing
Uni-processing
μP1
4
6
μP2
μP3
8
$
10
12
$
$
14
18
memory bus
Peripheral
AHB
CCB
AHB
CCB
AHB
CCB
16
SCU
(Snoop Control Unit)
IC
M0
M1
M2
2
22
coherent shared
Master AHB Interface
20
Memory

Fig. 1

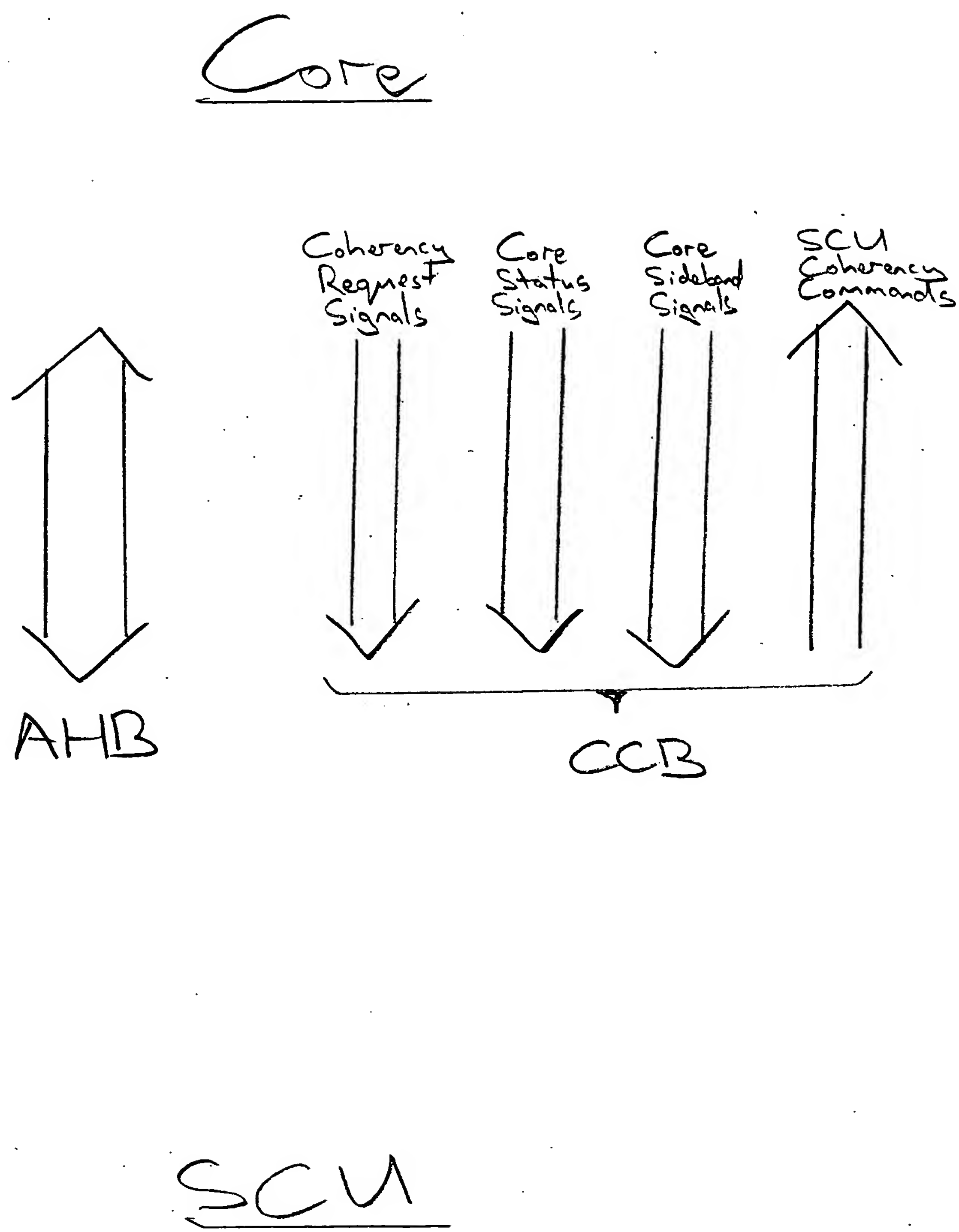
Core
Coherency Request Signals
Core Status Signals
Core Sideband Signals
SCU Coherency Commands
AHB
CCB
SCU

Fig. 2

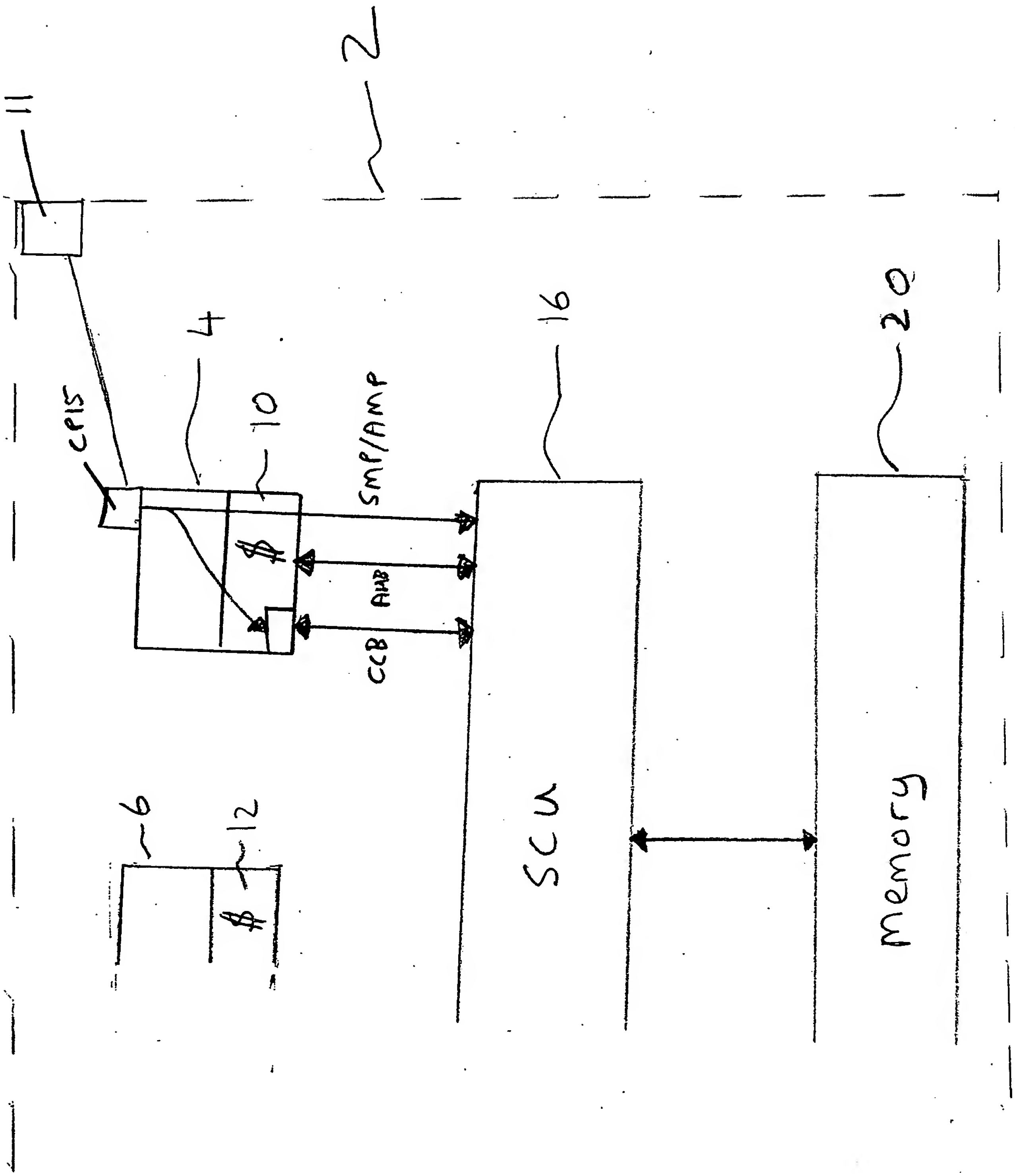

Figure 3

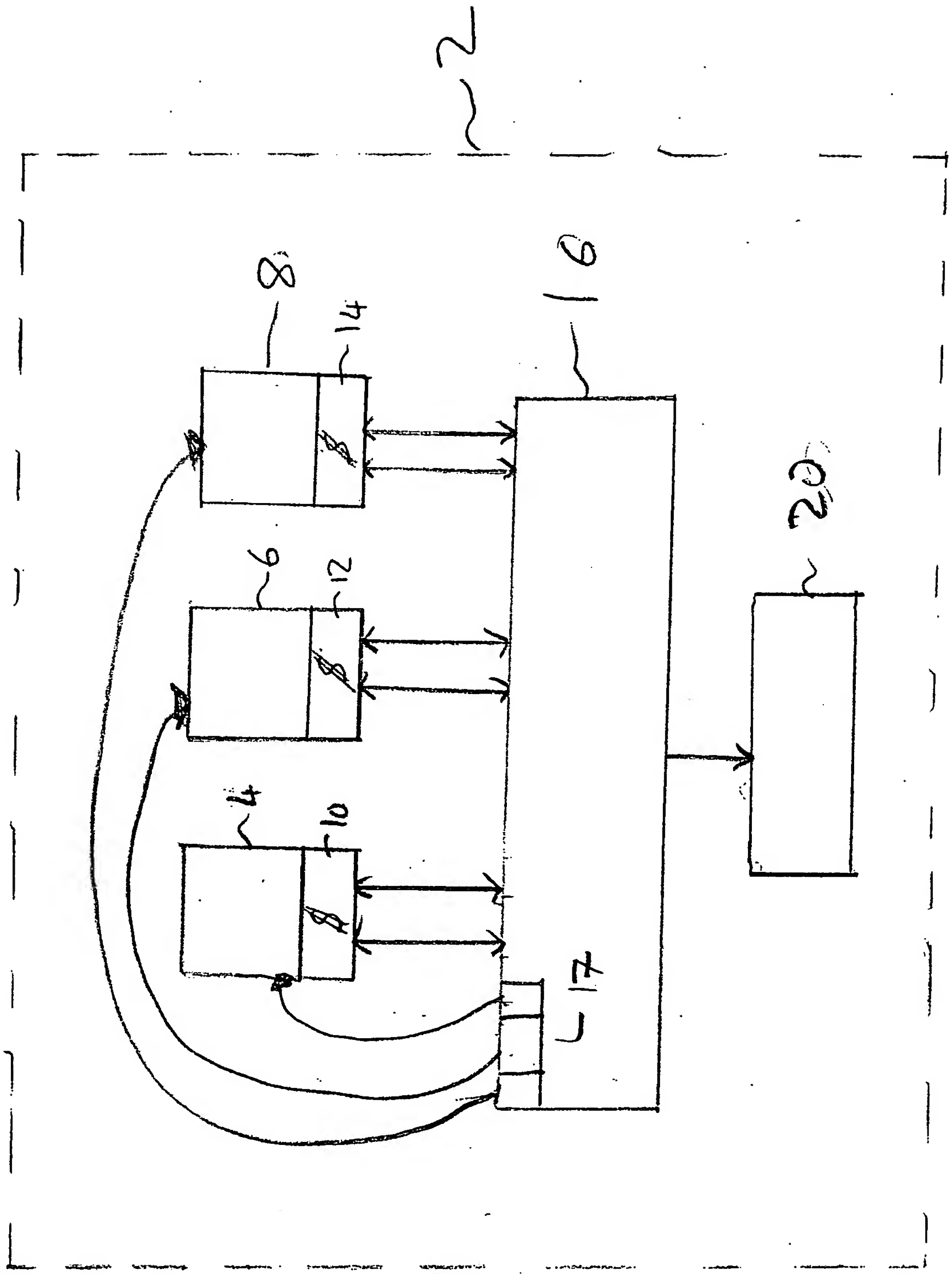

Figure 4

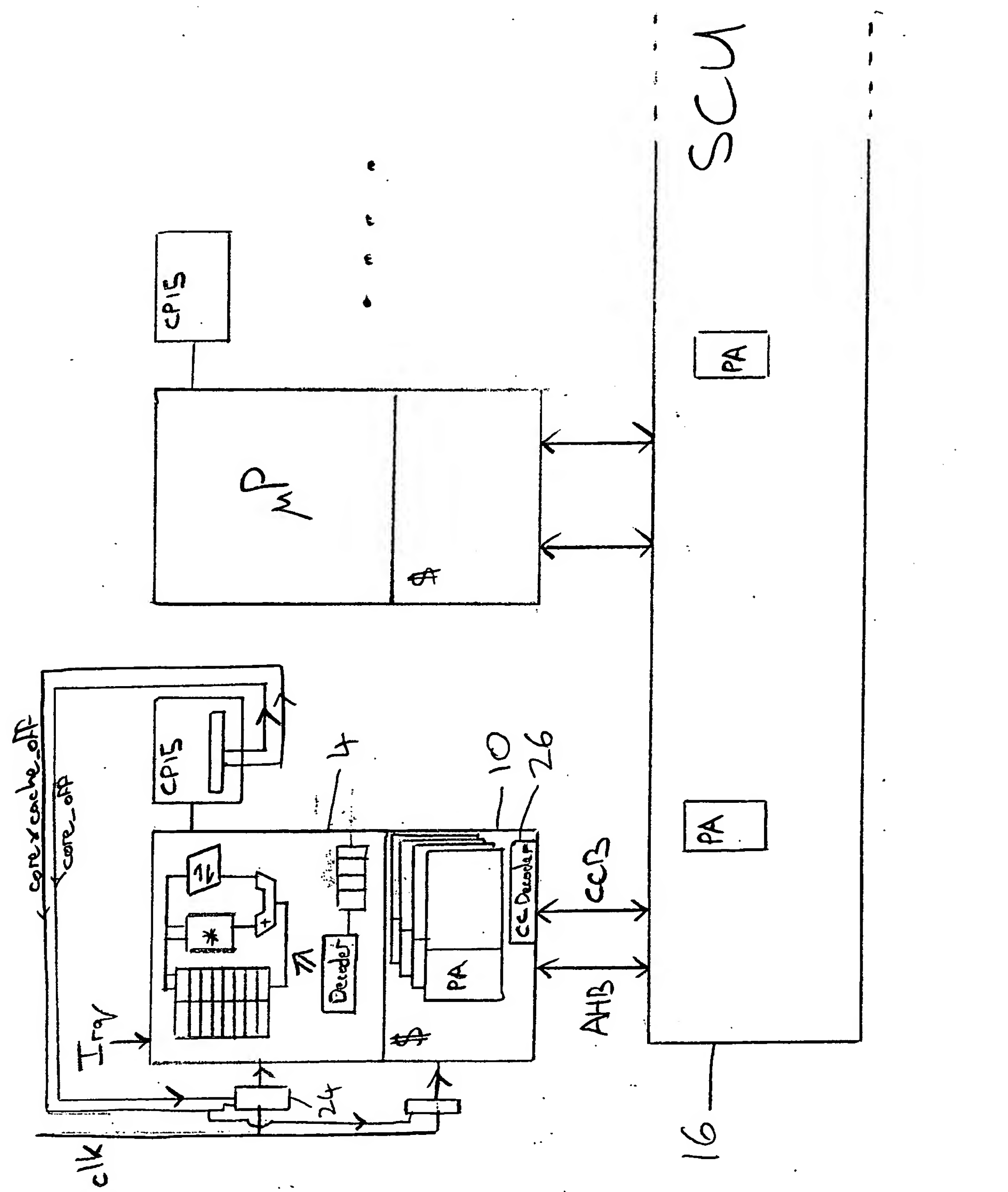
SCU
PA
PA
CP15
MP
$
CP15
4
10
26
Decoder
CC Decoder
PA
$
CCB
AHB
Irq
core&cache_off
core_off
24
clk
16

Fig. 5

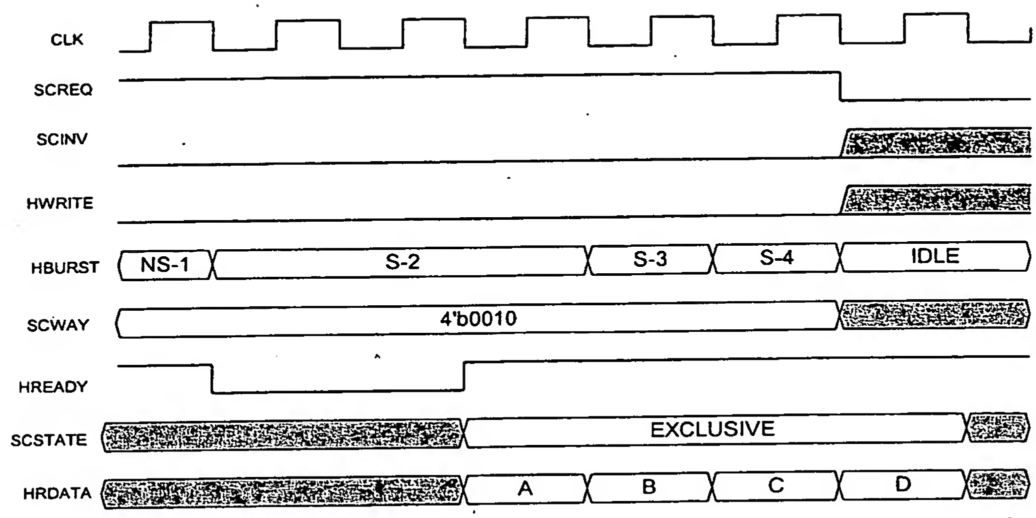
CLK
SCREQ
SCINV
HWRITE
HBURST
NS-1
S-2
S-3
S-4
IDLE
SCWAY
4'b0010
HREADY
SCSTATE
EXCLUSIVE
HRDATA
A
B
C
D

FIGURE 6

THIS PAGE BLANK (USPTO)

INVALIDATE ALL message

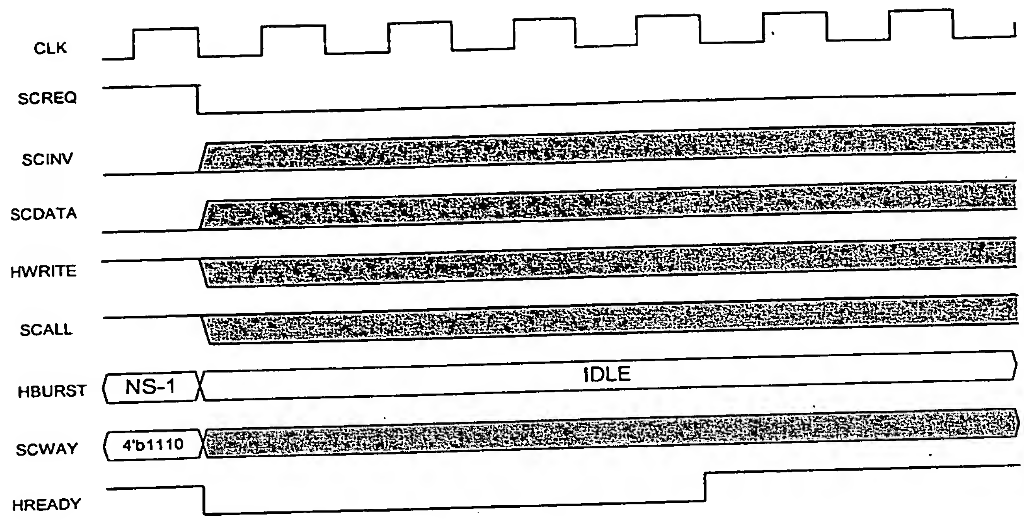

FIGURE 7

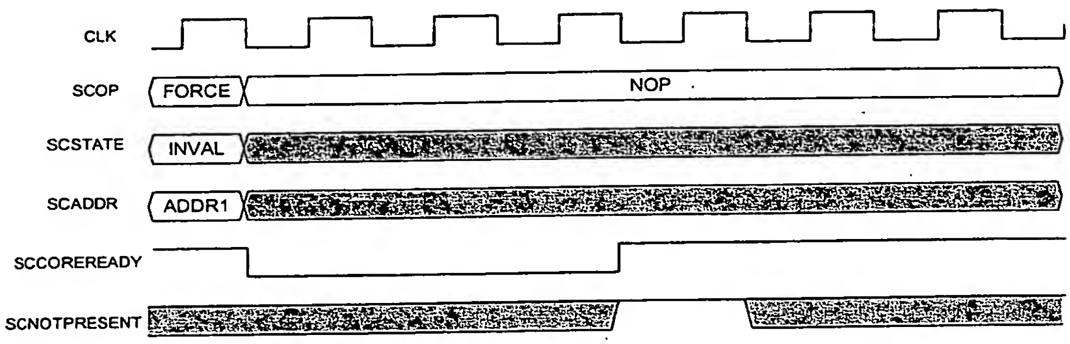
CLK
SCOP
FORCE
NOP
SCSTATE
INVAL
SCADDR
ADDR1
SCCOREREADY
SCNOTPRESENT

FIGURE 8

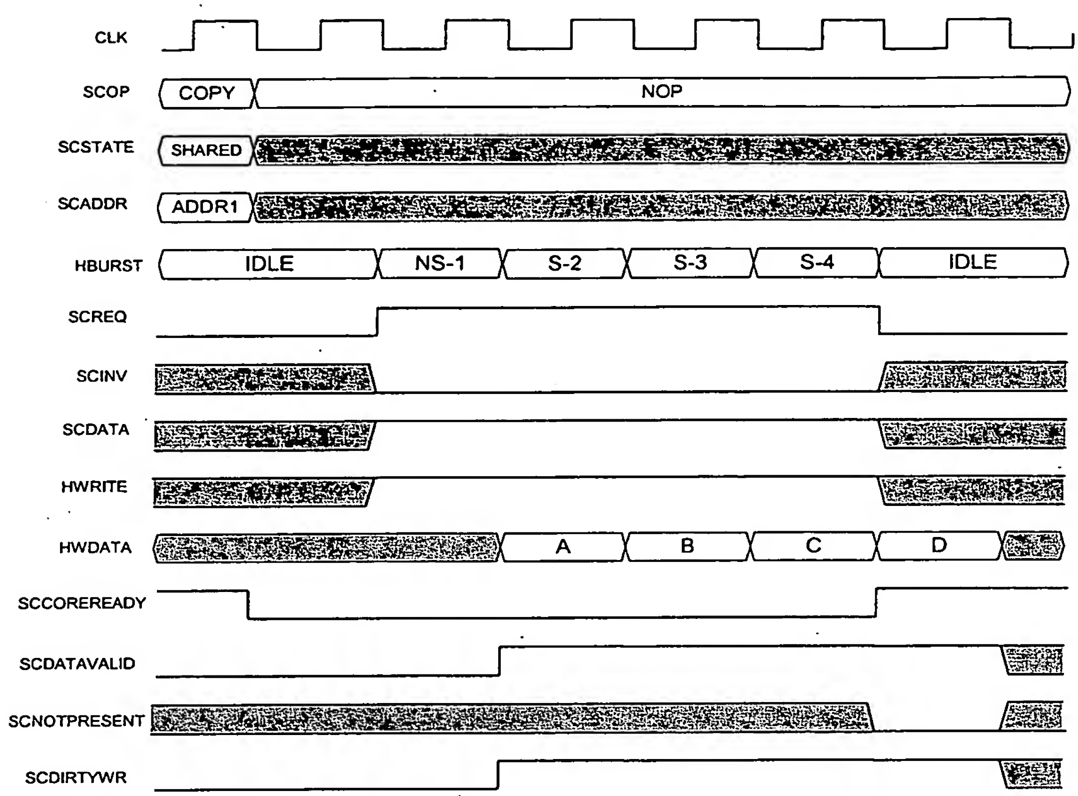
CLK
SCOP
COPY
NOP
SCSTATE
SHARED
SCADDR
ADDR1
HBURST
IDLE
NS-1
S-2
S-3
S-4
IDLE
SCREQ
SCINV
SCDATA
HWRITE
HWDATA
A
B
C
D
SCCOREREADY
SCDATAVALID
SCNOTPRESENT
SCDIRTYWR

FIGURE 9

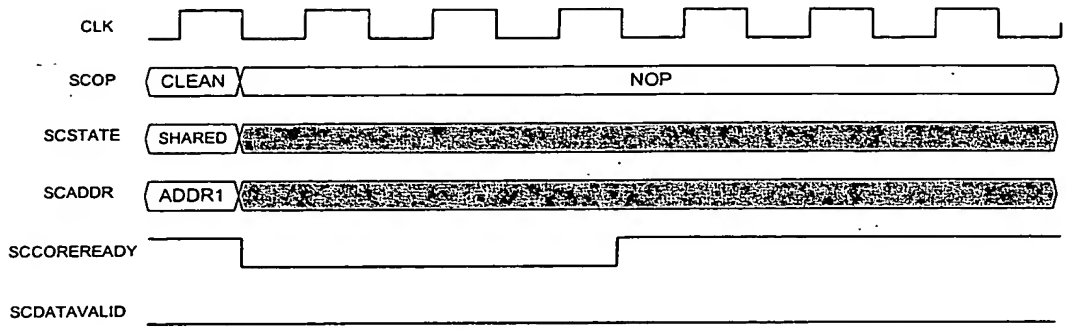
CLK
SCOP
CLEAN
NOP
SCSTATE
SHARED
SCADDR
ADDR1
SCCOREREADY
SCDATAVALID

FIGURE 10

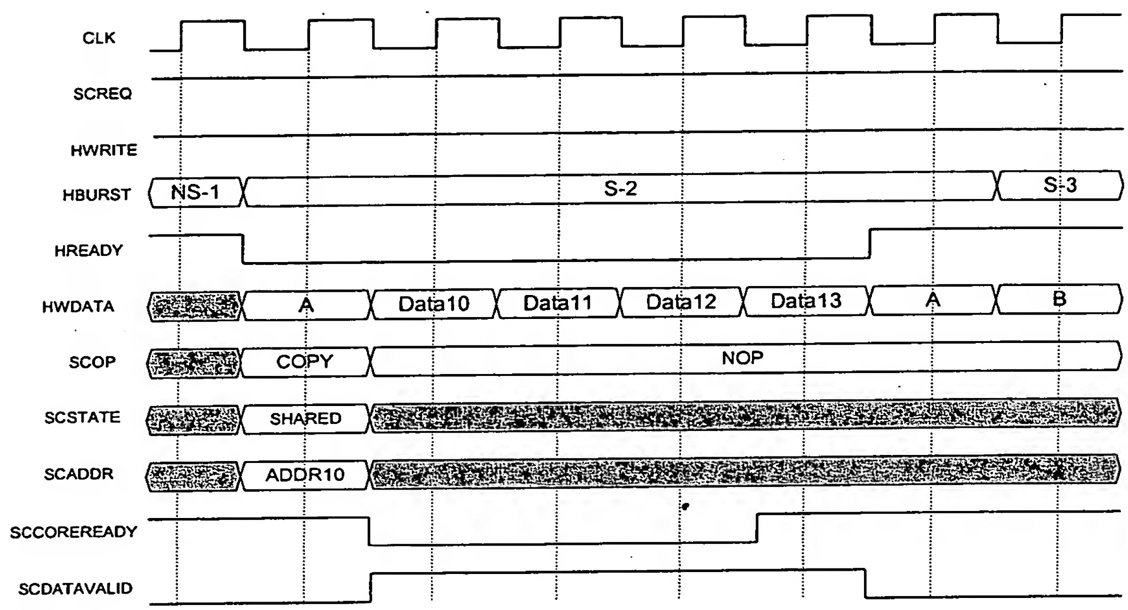
CLK
SCREQ
HWRITE
HBURST
NS-1
S-2
S-3
HREADY
HWDATA
A
Data10
Data11
Data12
Data13
A
B
SCOP
COPY
NOP
SCSTATE
SHARED
SCADDR
ADDR10
SCCOREREADY
SCDATAVALID

FIGURE 11